Monitoramento de logs da Plataforma HPCC SYSTEMS em Cloud

Nathália Geraldino Ribas
Universidade Federal de Santa Catarina - UFSC
ribas.nathalia@grad.ufsc.br

Alysson Oliveira
LexisNexis Risk Solutions
alysson.oliveira@lexisnexisrisk.com

Hugo Watanuki
LexisNexis Risk Solutions
hugo.watanuki@lexisnexisrisk.com

Patricia Plentz
Universidade Federal de Santa Catarina - UFSC
patricia.plentz@ufsc.br

ABSTRACT

Since log file monitoring is indispensable in the development and repair of an application, it is also of paramount importance to obtain a clear and direct interpretation when analyzing the files. Often due to the increase in data generated per second in current systems and the dispersion that cloud computing can cause in these files, this action can become quite complicated, having to analyze the file several times until the reported error is found. This task can take a long time, depending on the number of files, data, and components available on the system. In this study, the Elastic Stack log management tools were used: Elasticsearch, Filebeat, and Kibana for filtering, managing, centralizing, and viewing the logs. The tests were performed on LexisNexis' Big Data HPCC Systems platform. With the creation of interactive dashboards, the log visualizations and the search for errors were facilitated, which can help find solutions for reported errors.

PALAVRAS-CHAVE

COMPUTAÇÃO EM NUVEM, MONITORAMENTO DE LOG, ELASTIC STACK, HPCC SYSTEMS.

1 Introdução

A análise e monitoramento de *logs* ocupa uma posição muito importante no desenvolvimento de sistemas. Geralmente, quanto mais complexo for o sistema, mais importante se torna a análise e monitoramento de *log*. A partir do *log* podem ser realizadas a monitoração das condições de operação do sistema, identificação de comportamentos anormais, bem como o acionamento de notificações automáticas para equipes de suporte. A análise de *log* e o seu gerenciamento desempenham um papel importante na era de *big data*, especialmente, e espera-se que o avanço das tecnologias também permitam a mineração de dados a partir do *log* para serem realizadas análises estatísticas, análises de desempenho e taxa de sucesso, por exemplo [1].

Contudo, com o aumento das informações de *log* em massa, torna-se mais desafiador extrair as informações necessárias rapidamente. Além disso, o advento dos paradigmas de computação em nuvem e contêineres também geraram alguns desafios para a análise de *log*, uma vez que os mesmos agora além de serem gerados em massa, os *logs* podem estar dispersos em sistemas espalhados em diferentes infraestruturas. Por fim, caso a velocidade de consulta remota seja lenta, torna-se impraticável obter análise de *log* em tempo real [2]. Para lidar com esses desafios, novas tecnologias de busca e integração de *log* têm sido utilizadas [1].

Frente a esse cenário, o objetivo deste trabalho é fazer um estudo de caso quanto à utilização de ferramentas de gerenciamento de *logs* em uma plataforma de *Big Data* em um paradigma de contêineres em nuvem, e como a utilização de *dashboards* pode impactar na tomada de decisão diante de erros reportados no sistema. Para essa finalidade foi desenvolvido um ambiente experimental que utilizou a plataforma HPCC Systems (*High Performance Computing Cluster*) da empresa LexisNexis Risk Solutions e as ferramentas de coleta e visualização de *log* ElasticSearch e Kibana por meio da orquestração de contêineres via Kubernetes.

Este artigo está organizado da seguinte forma: na Seção 2 são descritos os principais conceitos relacionados. Na Seção 3 é discutido o principal problema, como ocorreu o desenvolvimento do *dashboard*, e as configurações principais. O resultado do projeto é apresentado na Seção 4. Por fim, a Seção 5 realça as principais contribuições do artigo e aponta direções para pesquisas futuras.

2 Conceitos Relacionados

2.1 Computação em nuvem

A computação em nuvem permite que armazenamento de dados, *softwares*, banco de dados e outros serviços sejam armazenados em um servidor remoto, por meio da *internet*, podendo ser acessados sob demanda. Assim, pode-se acessar os

dados que desejar, quando quiser e onde estiver. Isso, além de melhorar a agilidade da equipe, também barateia os custos, visto que a empresa precisa pagar apenas para o que usar e não precisará manter uma infraestrutura em seu local [3].

Uma pesquisa realizada pela MarketsandMarkets [4] mostra o crescimento da adoção desse serviço: estima-se que o mercado de computação em nuvem terá uma taxa de crescimento anual composta de 17.9% entre 2022 e 2027. A Figura 1 mostra o gráfico do aumento estimado por tipo de serviço e por ano até 2028. Observa-se um crescimento linear e consistente, reforçando a necessidade de pesquisas nessa área tecnológica.

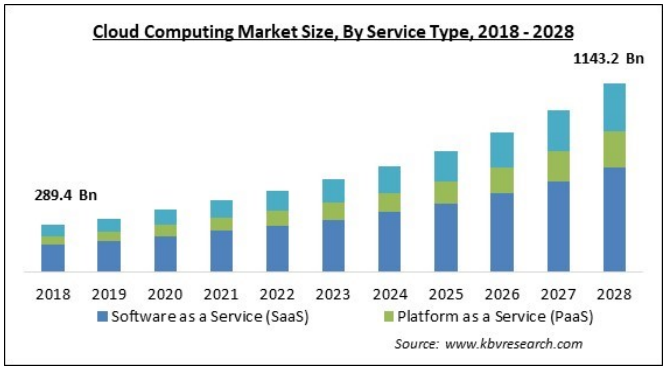

Figure 1: **Crescimento do mercado de computação em nuvem por tipo de serviço [5]**

A redução dos custos é um dos principais benefícios do uso da nuvem, já que não é preciso investir em grandes recursos para a infraestrutura. Além disso, também tem a redução de riscos para perda de dados — exemplos desses riscos são: quedas de energia elétrica, desastres naturais, falhas no *hardware* ou até mesmo roubo. Uma vez que a política dos *datacenters*, que implementam o conceito de nuvens, organiza os dados geograficamente distribuída, torna-se viável a recuperação quando políticas de tolerância a falhas e replicação são bem empregadas. Outros benefícios que também podem ser citados são: facilidade para o acesso dos serviços da organização e melhor gerenciamento dos recursos com uso sob demanda [6].

2.2 Logs

Os *logs* são documentos com descrições detalhadas, incluindo a data e hora exata, das ações processadas durante a execução de um *software* ou sistema operacional, bem como: mensagens de falhas, informes gerais, uso de memória e armazenamento, e outras informações que podem ser úteis para a análise. Com esse arquivo, é possível que o administrador da aplicação perceba onde o programa passou a apresentar falhas e por que ocorreu determinado erro, facilitando o monitoramento do ambiente e agilizando o alcance de soluções.

Os *logs* são muito importantes porque, além de ajudarem na solução de problemas de determinada aplicação, também ajudam no monitoramento geral do *software*. Isto é, com os *logs* é possível que visualize se o desempenho do sistema está favorável, se há alguma atividade maliciosa na aplicação e até quais foram as atividades realizadas pelos usuários.

2.3 Ferramentas de log

Ferramentas que organizam, filtram e disponibilizam uma versão mais visual dos detalhes contidos nos documentos podem ser de grande ajuda para realizar o monitoramento dos registros. Serão apresentados, neste artigo, componentes da Elastic Stack (Elasticsearch e Kibana) e o Elastic beat Filebeat.

a) Elasticsearch

O *Elasticsearch* é um mecanismo de busca e análise de dados *open source* [7]. Comumente utilizado para o gerenciamento de grandes quantidades de dados, podendo realizar a análise quase em tempo real (termo do inglês, *near real-time* [8]), além de ser rápido e escalável.

b) Filebeat

O *Filebeat* é responsável por centralizar os arquivos de *log*. Utilizando os locais de *log* que o administrador especificou, ele monitora, coleta e repassa os *logs* para o Elasticsearch ou Logstash [9].

c) Kibana

O *Kibana* é uma aplicação *web* que trabalha em conjunto com o Elasticsearch [10]. Com ele é possível fazer buscas avançadas e personalizadas nos arquivos de *logs*. Além disso, o Kibana é usado para fazer versões visuais dos relatórios, podendo criar gráficos dinâmicos dos mais diversos tipos que podem ser compartilhados e exportados.

A plataforma conta com a área de análise, onde possui a ferramenta para a criação de gráficos, como: gráficos de barras, pizza, tabelas, mapas e histogramas. Além disso, também contém as áreas de busca empresarial, de monitoramento de dados, e de segurança.

2.4 Ambientes conteinerizados

A conteinerização faz o empacotamento do código da aplicação. A solução foi criada com o objetivo de cessar as falhas que ocasionam quando uma aplicação é desenvolvida em uma plataforma X e é migrada para uma Y. Por exemplo: algumas aplicações são desenvolvidas no Linux e, quando tenta-se executá-las em um ambiente Windows, pode ocorrer uma falha de compatibilidade ou alguns outros problemas. Com a conteinerização, isso não ocorre [11].

Os *softwares* são desenvolvidos de maneira independente do sistema operacional, ou seja, todas as bibliotecas, configurações e dependências necessárias para o código rodar ficam embutidas no chamado contêiner — uma espécie de "pacote" que guarda a aplicação e suas configurações necessárias, como mostra a Figura 2, facilitando a migração entre plataformas. Assim, a aplicação se

torna universal e pode ser utilizada em vários sistemas operacionais e infraestruturas.

2.4.1 Benefícios da conteinerização

A portabilidade é um dos principais benefícios da conteinerização, dada por o aplicativo ser desenvolvido independente do sistema operacional hospedeiro (que está sendo executado na máquina principal), podendo ser utilizado em qualquer plataforma sem causar problemas de compatibilidade [11].

A característica do isolamento proporciona uma maior segurança para as aplicações, uma vez que elas não possuem contato entre si e nem com o sistema operacional hospedeiro. Além disso, os contêineres possuem um isolamento de falhas. Caso o contêiner de um aplicativo falhar, não afetará o funcionamento dos outros contêineres em execução. Inclusive, o time de desenvolvimento poderá corrigir erros sem causar inatividade em outras aplicações [11].

Outros pontos que podem ser citados são a agilidade e o tempo de início superior dos contêineres, já que não possui um sistema operacional para ser inicializado.

2.4.2 Diferenças entre contêineres e máquinas virtuais

Máquinas virtuais são como uma emulação de um computador. Cada máquina virtual criada possui especificações próprias, como: quantidade de memória RAM, processadores, quantidade de armazenamento e configurações de rede, e um sistema operacional específico. Com as máquinas virtuais, o usuário pode operar várias máquinas com sistemas operacionais diferentes em um único computador. O Hypervisor é um exemplo de *software* que faz o gerenciamento e criação das máquinas virtuais, alocando os recursos entre as diferentes máquinas [12].

Assim como os contêineres, as máquinas virtuais também fazem um isolamento das aplicações, porém, o isolamento das máquinas virtuais ocorre ao nível de *hardware*, enquanto nos contêineres ocorre ao nível de sistema operacional. A arquitetura dos contêineres possui apenas a aplicação e suas bibliotecas, sendo gerenciado por um gerenciador de contêineres. A figura 2 ilustra os componentes de cada uma das soluções e suas diferenças quanto à arquitetura [12].

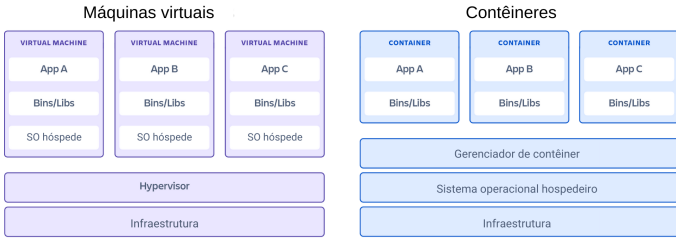

Figure 2: **Máquinas virtuais x Contêineres [13]**

Comparado com as máquinas virtuais, os contêineres são mais leves e possuem um menor tempo de inicialização, já que não precisam inicializar algum sistema operacional. Além disso, o problema com as migrações de aplicativos não é totalmente resolvido com aqueles criados em máquinas virtuais, podendo ocorrer falhas de compatibilidade quando transferidos para fora da máquina virtual.

2.5 HPCC SYSTEMS e seus componentes

Os computadores de alto desempenho (HPC, do inglês *High Performance Computing*) estão se tornando cada vez mais comuns em diversos setores. Essa tecnologia permite com que grandes quantidades de dados sejam processados rapidamente, resolvendo problemas que seriam muito pesados para computadores comuns ou que levariam muito tempo [14].

O HPCC SYSTEMS (*High Performance Computing Cluster*) é um sistema desenvolvido pela empresa LexisNexis que permite o armazenamento, análise e processamento de grandes quantidades de dados. A plataforma processa bilhões de dados por segundo utilizando tecnologia de processamento paralelo massivo, resolvendo diversos problemas de *big data*. HPCC possui componentes que podem ser divididos em 3 categorias: *clusters*, servidores e suas interfaces, os quais serão apresentados a seguir [15].

2.5.1 Clusters

Clusters são conexões compostas por dois ou mais computadores com o objetivo de melhorar o processamento de tarefas, sendo cada computador denominado "nó" ou "*node*". No contexto do HPCC, o sistema possui dois *clusters* principais: Thor e Roxie [15].

a) Thor

Thor, também conhecido como Refinaria de Dados, é responsável pelo processo ETL (*Extract, Transform and Load* - Extração, Transformação e Carga). O *cluster* usa uma abordagem de gestor/trabalhador (*manager/worker*) [16], onde possui um nó gestor e vários nós trabalhadores, como demonstrado na Figura 3. Enquanto os nós trabalhadores são responsáveis pelo processamento, armazenamento e aprimoramento dos dados, o nó gestor é encarregado de monitorar as tarefas dos trabalhadores. Por realizar tarefas de escrita e leitura, os nós, geralmente, necessitam de uma boa capacidade de armazenamento com um disco rápido.

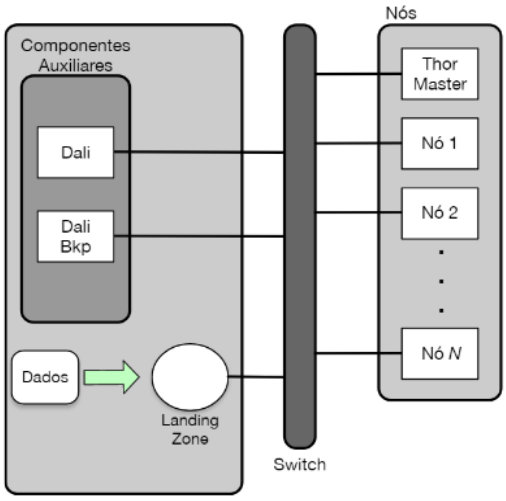

Figure 3: ***Cluster* Thor [15]**

b) Roxie

Roxie [17] é o motor de entrega rápida dos dados. O Roxie pode processar milhares de dados por segundo, os quais levariam alguns segundos ou minutos no Thor, dependendo da complexidade da *query*. As *queries*, ou seja, as requisições de informações ao banco de dados, são implementadas e pré-carregadas, assim elas estarão prontas quando forem solicitadas.

No que diz respeito ao funcionamento, no Roxie cada nó executa processos de servidores e agentes (Figura 4). O servidor recebe *queries* de serviços *Web* em formatos SOAP, HTTP ou HTTPS, e determina os nós e agentes que contêm os dados que foram pedidos. Uma vez recebido os dados dos agentes, o servidor agrupa-os e processa qualquer outra ação adicional, retornando a resposta para o cliente que solicitou a *query* [17].

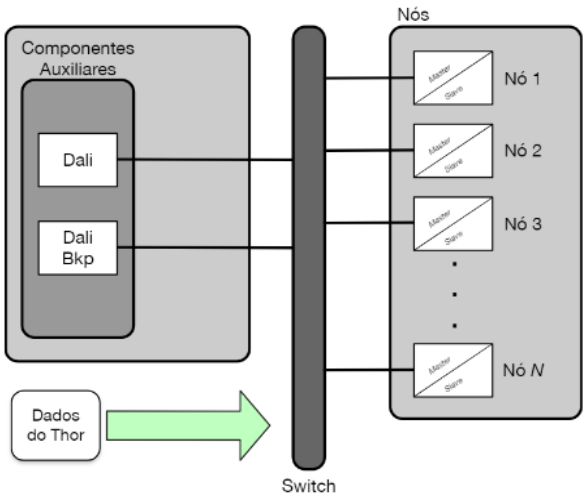

Figure 4: **Cluster Roxie [5]**

2.5.2 Servidores

Os servidores são responsáveis por criar uma ponte entre os *clusters* e o mundo exterior [18]. Cada servidor possui uma responsabilidade específica para ajudar a manter os dados e a conexão com os *clusters* mais otimizados.

a) Dali

Dali é o servidor que realiza o armazenamento de dados do sistema. Ele gerencia *workunits*, diretório de arquivos lógicos e serviços de objetos compartilhados. Além disso, Dali também é usado para configurar o ambiente, manter as filas de mensagens que cuidam da execução e do agendamento de tarefas, e aplicar a segurança do LDAP para escopos de dados e *workunits* [18].

b) Sasha

Sasha é um servidor organizador que trabalha em conjunto com Dali e trabalha independente de todos os outros componentes. Sua principal função é diminuir a carga e uso de recursos do Dali. Também é usado para arquivar *workunits*, remover *workunits* em cache e arquivos de recuperação do DFU [18].

c) DFU Server

DFU Server é responsável por controlar os processos de *spray* e *despray*, os quais movem dados para dentro e para fora do *cluster* Thor. Pode ser acessado através do ECL Watch, pela linha de comando do DFU e pode ser utilizado em códigos ECL através de bibliotecas [18].

d) ECLCC Server

ECLCC Server é o compilador dos códigos ECL, responsável por transformar os códigos em C++, o qual é compilado e executado. O servidor é utilizado para compilar as *workunits* do Thor e Roxie, para verificação de sintaxe dos códigos ECL, e é encarregado de iniciar um processo de ECL Agent quando uma *workunit* é executada [18].

e) ECL Agent (hThor)

A principal função do ECL Agent é transferir cada tarefa para seu respectivo *cluster*. Porém, quando uma tarefa é simples demais para ser executada pelo Thor, o hThor entra em ação agindo como um *cluster* de nó único [18].

f) ESP Server

ESP Server é um *framework* que possibilita a conexão e comunicação entre múltiplos serviços. Alguns dos serviços conectados no ESP são: WsECL, uma interface *Web* para publicação de *queries* do Thor e Roxie; e o ECL Watch, uma plataforma *Web* para realizar tarefas e gerenciar *clusters*. O ESP Server suporta os protocolos XML e JSON [18].

2.5.3 Interfaces

a) ECL IDE

O ECL IDE é o ambiente usado para a criação dos códigos ECL. Nele é possível acessar repositórios, executar e depurar códigos,

e outras funcionalidades de monitoramento e gerenciamento de tarefas e *clusters* [18].

b) ECL Watch

ECL Watch é uma plataforma *Web* de gerenciamento de arquivos e monitoramento de *clusters*. Por meio da interface é possível acessar e executar *workunits* e consultar seus gráficos, realizar processos de *spray* e *despray*, visualizar os *status* dos sistemas e muitas outras funcionalidades [18].

3 Desenvolvimento do Dashboard para Análise de Logs

3.1 Discussão do Problema

Atualmente, quando um erro é reportado para o administrador do sistema, o mesmo deve ler, em alguns casos, centenas de linhas de arquivos de *log* para perceber o que está ocorrendo e o porquê.

O fluxograma abaixo, na Figura 5, define como o processo de procura de erros em arquivos de *log* ocorre atualmente na empresa: quando um erro é reportado para o administrador do sistema, o mesmo deve primeiramente investigar em quais dos componentes a mensagem de erro pertence, para então poder procurar nos registros do componente a mensagem de erro ou o ID do processo que falhou. Esse processo deve se repetir até o administrador encontrar o erro reportado, e o tempo que demora para ter êxito depende diretamente da quantidade de nós que o *cluster* possui.

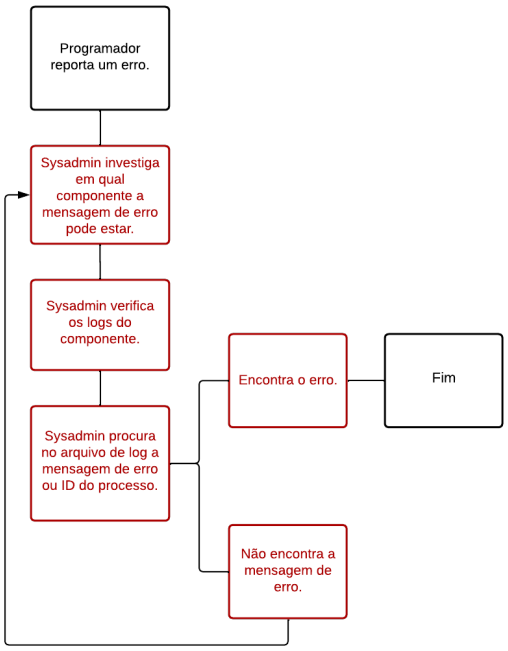

Figure 5: **Fluxograma do processo de procura por erros**

Esse processo é bastante repetitivo, dependendo da quantidade de nodos e componentes que o processo utiliza, podendo tornar a procura cansativa e bastante difícil.

Por conta desses motivos, o desenvolvimento de *dashboards* para a visualização dos *logs* foi encontrado como uma solução para tornar a procura por erros de um administrador de sistema, ou até mesmo de um programador, mais fluída e facilitada.

3.2 Processo do desenvolvimento

A pesquisa foi desenvolvida em torno da análise de logs gerados em consequência dos testes na plataforma HPCC Systems da LexisNexis. Foram forçados alguns erros para visualizar a forma que eles eram descritos na plataforma e no ambiente de visualização de *logs* (Kibana).

O objetivo principal era perceber quais informações estavam sendo transferidas para o ambiente de visualização e como que nos permite tornar os dados mais intuitivos por meio da criação de *dashboards* que podem auxiliar na monitoração.

A ferramenta utilizada para a criação dos *dashboards* foi a Kibana, da Elastic Stack. A *stack* possibilita a extração e centralização dos arquivos de *log*, sendo o Kibana a interface que nos permite visualizar, filtrar e gerenciar da melhor forma.

Os *dashboards* foram criados a partir de filtros específicos, como, por exemplo: id da *workunit* (processo submetido ao sistema), mensagens de erros ou ações que o sistema tomou durante o processo, tempo (dia/hora), e quantidade total de processos. Os filtros foram criados utilizando a KQL, linguagem do Kibana para criação de filtros, e em dois modelos que estão sendo mostrados na Figura 6.

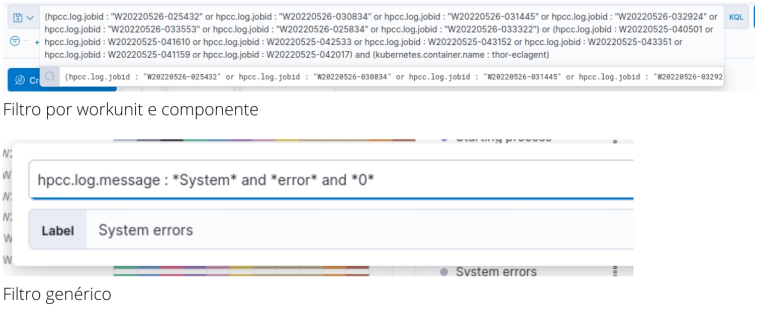

Figure 6: **Tipos de filtros utilizados**

3.3 Ambientes e configurações

A fim de verificar a funcionalidade do *dashboard*, alguns testes foram realizados. Na primeira sessão de testes, as aplicações foram inicializadas utilizando uma máquina local com processador Intel Core i5-1035G1 CPU @ 1.00GHz × 8, com 8 GB de memória RAM, sistema operacional Ubuntu 20.04 com arquitetura de 64 bits.

Os testes foram realizados na plataforma HPCC Systems versão 8.2.2, utilizando o Elastic em sua versão 7.17.1, ambos estando em

um contêiner Docker padrão. Os detalhes de funcionamento e as características da plataforma HPCC, assim como do Elastic estão expostas na seção 2 deste artigo.

4 Resultados

O *dashboard* desenvolvido contém cinco componentes (Figura 7): uma tabela com todas as *workunits* processadas, contagens de ocorrências no geral, por *workunit*, e por dia, e os processos realizados pelo sistema durante o tempo filtrado. A necessidade desses cinco componentes foram percebidas à medida que os testes estavam sendo submetidos no ambiente da empresa. Melhorias e acréscimos ainda estão sendo discutidos, para tornar o processo cada vez mais facilitado.

Figure 7: **Visão geral do dashboard**

Os componentes podem ser visualizados com mais detalhes nas figuras 8 a 13. A Figura 8 mostra a lista de *workunits* com o número de ocorrências em cada uma delas. Essa tabela pode ser útil para os programadores que queiram ter a visualização de apenas suas *workunits*, uma vez que o Kibana permite a filtragem por qualquer um dos elementos da tabela.

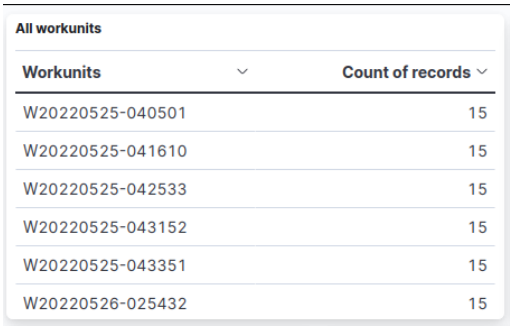

Figure 8: **Lista de workunits**

A Figura 9 apresenta um componente com os processos mais comuns encontrados durante o teste e a quantidade de vezes que ele ocorreu em cada uma das *workunits*. É possível, também, filtrar por processo e visualizar quais *workunits* passaram por aquele determinado processo filtrado.

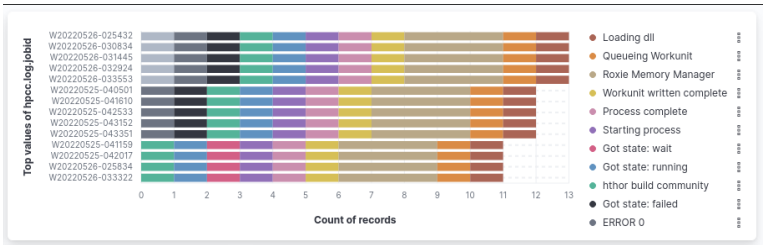

Figure 9: **Contagem de processos por *workunit***

O número de ocorrências dos processos gerais e o número de ocorrências por dia são mostrados no *dashboard*, conforme mostram as Figuras 10 e 11. Ambas visualizações são necessárias para o administrador do sistema verificar quantos processos foram executados em determinado dia e quantos foram executados no período de tempo solicitado.

Figure 10: **Processos gerais** Figura 11: **Filtro por dia**

Na Figura 12 apresentam-se os processos durante um período de tempo. Essa informação pode ser importante para acompanhar os processos na ordem que foram realizados e como se seguiu.

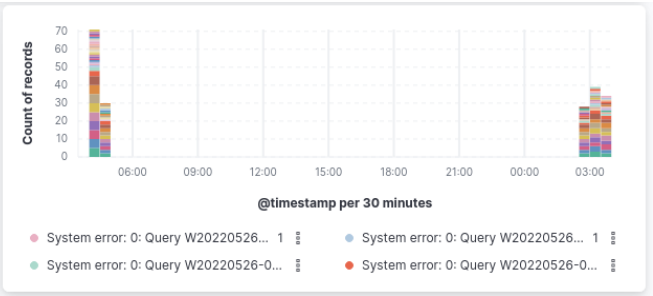

Figure 12: **Processos durante tempo**

Uma das grandes vantagens encontradas no Kibana é a possibilidade de filtragem por diversos parâmetros: *pod*, mensagens, tempo e qualquer outro tipo de componente presente na estrutura das mensagens do Kibana. Com a criação do *dashboard* proposto neste artigo e com a utilização da plataforma, espera-se facilitar o processo de procura por erros em arquivos de *log*. Pode-se observar que foram criadas várias formas de visualização das informações que fazem parte do

HPCC Systems. Estas formas de visualização facilitam e aceleram a rotina dos desenvolvedores e analistas do sistema.

5 Conclusão

Este artigo apresentou o desenvolvimento de um *dashboard* para monitoramento de *logs* para computação de alto desempenho em sistemas distribuídos. Utilizamos a plataforma HPCC Systems, Kibana e dados reais usados no dia-a-dia da empresa.

A análise e a monitoração dos arquivos de *log* podem se tornar bastante eficiente com o uso da Elastic Stack, principalmente por conta das possibilidades de centralização de *log*, filtragem de mensagens de erros e componentes, e a criação de *dashboards* que proporcionam uma visualização do que está acontecendo no sistema. Essas características podem facilitar a procura por erros e os seus motivos, processos gerais e o *status* atual de cada um dos componentes.

O *dashboard* desenvolvido neste projeto está em fase de testes na empresa e será usado para auxiliar os gerentes do sistema para encontrar erros nos arquivos. Esse tipo de ferramenta acelera as entregas das equipes e facilita o ciclo de detecção e correção de erros nos arquivos de *logs*.

REFERÊNCIAS

[1] Lei, X., Wang, Z., & He, Y. (2015, October). Log Real-time Management Scheme Based on LEK. In 2015 2nd International Workshop on Materials Engineering and Computer Sciences (pp. 206-209). Atlantis Press.

[2] Singh, S. (2016). Cluster-level Logging of Containers with Containers: Logging Challenges of Container-Based Cloud Deployments. Queue, 14(3), 83-106.

[3] Zettler, K. What is cloud computing? An overview of the cloud. Atlassian. Disponível: https://www.atlassian.com/continuous-delivery/principles/cloud-computing. Acesso: 15 de março de 2022.

[4] MarketsandMarkets. Cloud Computing Market by Service Model (Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS)), Deployment Model (Public and Private), Organization Size, Vertical, and Region - Global Forecast to 2026. MarketsandMarkets, 2021. Disponível em: https://www.marketsandmarkets.com/Market-Reports/cloud-computing-market-234.html. Acesso: 16 de março de 2022.

[5] Kbv Research. Global Cloud Computing Market Size, Share & Industry Trends Analysis Report By Service Type, By Deployment, By Enterprise Size (Large Enterprises and Small & Medium Enterprises), By End-use, By Regional Outlook and Forecast, 2022 - 2028. Kbv Research, 2022. Disponível: https://www.kbvresearch.com/cloud-computing-market/. Acesso: 08 de fevereiro de 2022.

[6] Jones, E. 11 Benefits of Cloud Computing in 2022. Kinsta, 2021. Disponível em: https://kinsta.com/blog/benefits-of-cloud-computing. Acesso: 16 de março de 2022.

[7] Elastic. What is ElasticSearch?. Elastic. Disponível em: https://www.elastic.co/guide/en/elasticsearch/reference/current/elasticsearch-intro.html. Acesso: 16 de março de 2022.

[8] Elastic. Near real-time search. ELASTIC. Disponível: https://www.elastic.co/guide/en/elasticsearch/reference/current/near-real-time.html. Acesso: 08 de fevereiro de 2023.

[9] Elastic. Filebeat Overview. Elastic. Disponível em: https://www.elastic.co/guide/en/beats/filebeat/current/filebeat-overview.html. Acesso: 16 de março de 2022.

[10] Elastic. Kibana-your window into ElasticSearch. Elastic. Disponível em: https://www.elastic.co/guide/en/kibana/current/introduction.html. Acesso: 16 de março de 2022.

[11] IBM Cloud Education. Containerization. IBM, 2021. Disponível: https://www.ibm.com/cloud/learn/containerization. Acesso: 11 de maio de 2022.

[12] IBM Cloud Team. Containers vs. Virtual Machines (VMs): What's the Difference?. IBM, 2021. Disponível: https://www.ibm.com/cloud/blog/containers-vs-vms. Acesso: 13 de maio de 2022.

[13] Buchanan, I. Containers vs Virtual Machines. Atlassian. Disponível: https://www.atlassian.com/microservices/cloud-computing/containers-vs-vms. Acesso: 13 de maio de 2022.

[14] Oracle. What is High Performance Computing (HPC)?. Oracle. Disponível: https://www.oracle.com/cloud/hpc/what-is-high-performance-computing. Acesso: 11 de Outubro de 2021.

[15] Middleton, A. M.; Chala, A., White Paper HPCC Systems: Introduction to HPCC (High Performance Computing Cluster). Maio de 2011.

[16] HPCC Systems. Introduction to HPCC Systems Open Source Big Data Platform. HPCC Systems, 2022. Disponível: https://cdn.hpccsystems.com/whitepapers/wp_introduction_HPCC.pdf. Acesso: 08 de fevereiro de 2022.

[17] HPCC Systems. Roxie: The Rapid Data Delivery Engine. HPCC Systems, 2016. Disponível: https://cdn.hpccsystems.com/releases/CE-Candidate-6.2.0/docs/RoxieReference-6.2.0-1.pdf. Acesso: 9 de Outubro de 2021.

[17] HPCC Systems. Guia do Administrador do HPCC System. HPCC Systems, 2021. Disponível: https://cdn.hpccsystems.com/releases/CE-Candidate-8.4.0/docs/PT_BR/HPCCSystemAdministratorsGuide_PT_BR-8.4.0-1.pdf. Acesso: 30 de Setembro de 2021.